Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 2, No. 1, 2022

A Network Intrusion Detection Model Based on Principal Component Analysis and Random Forest

Le Yang [1], Hua Chen[2] *

[1] Foshan Network Security Emergency Command Center, Foshan, 528000, China
[2] Foshan Human Resources Public Service Center, Foshan, 528000, China
* **Corresponding author**: Hua Chen (Email: chenh393@mail2.sysu.edu.cn)

Abstract: Network intrusion data has the characteristics of high dimension, nonlinearity and redundancy. To solve the problem of low detection rate of traditional dimensionality reduction and detection methods, a network intrusion detection method based on principal component analysis combined with random forest is proposed. Firstly, the data dimension of network intrusion is reduced by principal component analysis to eliminate redundant information between data, and then the processed data is classified and trained by using random forest classifier. The algorithm is verified by the network intrusion NSL_KDD dataset. The experimental results show that compared with other network intrusion detection methods, the method has fast learning speed, high detection accuracy, low false negative rate and low false positive rate. An efficient, real-time and good network intrusion detection method.

Keywords: Network security; Intrusion detection; Principal component analysis; Random forest.

1. Introduction

With the development and application of information technologies such as cloud computing and big data, the scale of various server cluster networks is becoming more and more large and complex, and at the same time, it is also facing more and more network security threats [1]. Network intrusion detection is one of the effective means to improve network security. It judges normal behavior or abnormal behavior according to network traffic data or host data, which can be abstracted into classification behavior, and machine learning has a powerful ability to solve classification problems, so many Research attempts to use machine learning algorithms in intrusion detection models [2].

Among various intrusion detection methods based on machine learning models, the method based on random forest model has good results in terms of training time, false alarm rate, and detection ability of unknown attacks. Reference [3] uses One-R fast attribute to solve the problem of inefficiency caused by excessive randomness in the selection of attributes of the random forest model when encountering high-dimensional data, and achieved good spatiotemporal performance and low false positive and false negative rates in the experiment. Literature [4] proposes a method of using KNN to delete outlier data and then combining with multi-level random forest to detect network attacks, which can effectively detect Probe, U2R, R2L and other attacks on the KDD CUP99 dataset.

Some other methods combined with artificial intelligence are also used in network intrusion detection. Reference [5] proposes a PCA-BP neural network intrusion detection method. By reducing the dimension of data features and correcting weights, the shortcomings of slow convergence of BP neural network are improved and the detection performance is improved. Literature [6] proposes an intrusion detection algorithm based on PCA and SVM (Support Vector Machines), but the accuracy is low for individual attack types.

This paper proposes an intrusion detection algorithm based on PCA and random forest classification. The data is first reduced by PCA and then classified by random forest, and the effectiveness of the algorithm is verified by experiments.

2. Decision tree and random forest

2.1. Decision tree

A decision tree is a classifier model and the basis for forming random forests. Common decision tree generation algorithms mainly include ID3 [7], C4.5 [8], CART [9], etc. Among them, ID3 algorithm is the first influential decision tree generation algorithm, and the latter two algorithms are in its On the basis of optimization or borrowing its idea. The ID3 algorithm introduces the concept of information entropy in information theory and defines the information gain of feature attributes.

Assuming that a total sample set D can be divided into m different categories according to the target attribute, then the information entropy of this sample is:

$$H(D) = \sum_{i=1}^{m} -p_i \log_2 p_i \tag{1}$$

Where p_i is the proportion of the ith type of subsample set D_i in the total sample set D , or the probability that D_i appears.

The information gain brought by an attribute to the whole is equal to the difference between the total information entropy and the residual information entropy after the attribute is selected. Assuming that the information gain of attribute A is to be obtained, and attribute A has k different values, the total sample should be divided into several sub samples $(D_1, D_1, \cdots, D_k)$ according to the value of attribute A, and the entropy of each sub sample should be calculated respectively, and then the residual information entropy can be obtained by weighted summation.

$$R(A) = \sum_{i=1}^{k} \frac{len(D_i)}{len(D)} H(D_i) \tag{2}$$

Where len () represents the size of the sample set in parentheses. According to the total information entropy and residual information entropy, the information gain of attribute A is:

$$G(A) = H(D) - R(A) \tag{3}$$

According to the information entropy theory, when the probability of occurrence of each type of sub-sample in the total sample is the same, that is, for any i, there is $p_i = 1/m$, the maximum information entropy is $\log_2^m$. However, if the probability of occurrence of one type of subsample is much higher than that of other types of subsample, the smaller the information entropy is, and when there is only one type of subsample, the minimum information entropy is 0. Therefore, the key idea of ID3 algorithm is to minimize the residual information entropy after classification according to certain attributes, so that the purity of each subsample after division is higher. Every time ID3 algorithm selects the feature attributes used for splitting, it needs to calculate the information gain of all the feature attributes, select the feature attribute with the largest information gain, divide the sample into several sub samples according to this feature attribute, and then repeat the process until all the samples belong to the same category, or most of the samples belong to the same category.

2.2. Random forest

Because decision trees are easy to over fit when facing samples with higher dimensions, most practical applications use integrated models based on decision trees, in which random forest is a classifier composed of multiple independent decision trees. Random forest has the advantages of strong generalization ability, fast training speed, can deal with high-dimensional data and does not need feature selection. It has been widely used in many classification problems.

The training process of random forest is actually to build several decision trees. Until each decision tree is built, the random forest will be trained. The training samples of each decision tree are sampled from the whole training samples by Bootstrap method, and the feature attributes used for classification of each decision tree are randomly selected from all feature attributes, usually the number of selected attributes is less than the total number of feature attributes.

During the test of random forest, for any test sample, each decision tree will independently judge the category of the sample, and finally the classification results of the entire random forest will be obtained by voting the classification results of all decision trees. Suppose that for a test sample X, the output of the kth decision tree is

$$f_k(X) = i \tag{4}$$

Where i represents the serial number of a category, the set of serial numbers of the decision tree output as i is:

$$S_i = \{k \mid f_k(X) = i, k = 1,2,\cdots,K\} \tag{5}$$

Where *K* represents the number of decision trees. So, the output of the whole random forest is:

$$f_k(X) = \underset{i=1,2,\cdots,m}{\arg\max}(len(S_i)) \tag{6}$$

Where $\underset{i}{\arg\max}()$ represents the largest expression value in the parentheses in all i, and $len()$ represents the size of the set in the parentheses.

2.3. Principal Component Analysis

Principal Component Analysis is a data analysis method proposed by K. Pearson more than a century ago. Principal component analysis is also a common multi-attribute analysis method. Its core idea is data dimension reduction. The feature of principal component analysis is very meaningful. When there are many analysis indicators and the process is complex, principal component analysis can simplify the analysis process and reduce the amount of calculation. Because PCA has the advantages of objectivity, simple calculation and convenient application, it has been widely used in mathematical modeling, mathematical analysis and other disciplines

The detailed steps of principal component analysis are as follows:

Establish evaluation matrix

Suppose the number of network attack sample is m, and the number of data characteristic parameters (evaluation parameters) is n, then the evaluation matrix is:

$$\mathrm{X}=\begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} = \left[x_{ij}\right]_{m\times n} \quad (i=1,2,...,m;\ j=1,2,...,n) \tag{7}$$

Where x_{ij} represents the evaluation value of the *jth* parameter of the *ith* sample.

Normalize evaluation matrix

The purpose of normalization is to eliminate the dimensional influence between parameters, so as to solve the comparability between data parameter indicators. After data normalization, the original data are in the same order of magnitude, which is suitable for comprehensive comparison and evaluation. The standardized formula is as follows:

$$z_{ij} = \frac{x_{ij} - \overline{x_j}}{\sigma_j} \tag{8}$$

$$\overline{x_j} = \frac{1}{m}\sum_{i=1}^{m} x_{ij} \tag{9}$$

$$\sigma_j = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(x_{ij} - \overline{x_j}\right)^2} \tag{10}$$

Where $\overline{x_j}$ is the average value of the *jth* parameter and σ_j is the variance of the *jth* parameter? The standardized matrix is:

$$Z = \left[z_{ij}\right]_{m\times n} (i=1,2,...,m;\ j=1,2,...,n) \tag{11}$$

Calculate the correlation coefficient matrix R.

$$R = \left[r_{kj}\right]_{n\times n} (k, j = 1,2,...,n) \tag{12}$$

Solve the eigenvalues and eigenvectors of the correlation coefficient matrix, and determine the principal components. Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ are the eigenvalues of the

correlation coefficient matrix and $L_{g1}, L_{g2}, \cdots, L_{gi}$ are the corresponding eigenvectors. Then the solution of the principal component is as follows:

$$F_i = \sum_{j=1}^{n} L_{gi} z_{ij} (i = 1, 2, \cdots, n) \quad (13)$$

Determine the number of principal components. In order to minimize workload and information loss, only the first k principal components are retained. The value of k can be determined by the cumulative contribution rate $\alpha(k)$, and the criteria are as follows:

$$\alpha(g) = \lambda_g / \sum_{g=1}^{n} \lambda_g \quad (14)$$

$$\alpha(k) = \sum_{g=1}^{k} \alpha_g = \sum_{g=1}^{k} (\lambda_g / \sum_{g=1}^{n} \lambda_g) \quad (15)$$

In equation (14), $\alpha(g)$ refers to the contribution rate of each principal component. In general, the greater the value of principal component contribution rate $\alpha(g)$, the richer the information of the original variables contained in the principal component. Generally, k principal components with cumulative contribution rate higher than 80% and eigenvalue greater than 1 are taken as the final selected principal components.

After determining the principal component fraction, the eigenvector corresponding to the eigenvalues of each principal component is calculated according to the eigenvalues of the correlation coefficient matrix, and the impact of each index on the principal component can be known according to the descending order of the eigenvector coefficients. The larger the value of the coefficient, the greater the influence of the index on the principal component. If the coefficient value is less than 0.1, the influence on the principal component is negligible.

According to the above steps, a complete indicator parameter system can be established, which can fully reflect the information contained in each indicator parameter, eliminate the correlation between each indicator parameter, and facilitate further data analysis.

2.4. Intrusion Detection Algorithm Based on PCA and Random Forest Classification

PCA algorithm has good data processing ability. It can reduce the dimension of a large number of data to reduce the amount of data while retaining the main information in the data, and can remove noise such as outliers in the data. But for a group of data, the algorithm itself cannot give any useful information, and the random forest classifier can effectively classify the data and can effectively prevent the data from over fitting. Therefore, according to the characteristics of PCA and random forest classification, this paper proposes an intrusion detection algorithm based on PCA and random forest classification, which combines PCA and random forest classification to improve the accuracy of classification and reduce the false alarm rate of the algorithm. The training steps of intrusion detection algorithm are as follows:

Select the number of maximum characteristic values (k value) to be retained according to the properties of the original data.

PCA is used to reduce the dimension of the data (retain the first k features with the largest eigenvalue), and remove the noise.

According to the data characteristics after preprocessing, the forest density (the number of decision trees) of the random forest classifier is selected.

The random forest classifier is used to train the data to get a classifier that can be used for intrusion detection.

Select different k values and forest densities to repeat the above steps, compare the impact of different k values and forest densities on the classifier, and select the best classification result as the final classifier.

The training flow chart of the intrusion detection algorithm proposed in this paper is shown in Figure 1. Point value PCA classifier is selected for initial data, and forest density is selected for random forest classification. It is worth mentioning that, because the characteristics of different data sets are very different, the ideal k values and forest density values under different data sets are often very different, and there is no universally applicable optimal solution. For general data sets, if the k value is too large, the PCA feature reduction and denoising effect cannot be well exerted; if the value of k is too small, the data will lose most of its original information. In addition, the data distribution of some datasets is too sparse. For these datasets, the ideal k value is often small; for data sets with very dense data distribution, the ideal k value is often large, or even nearly the same dimension as the original data set.

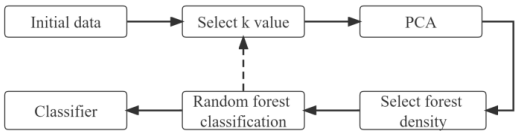

Figure 1. Flow chart of intrusion detection algorithm

There are also corresponding judgments on the forest density of random forests. If the number of decision trees in the forest is too small, the over fitting phenomenon of data cannot be eliminated well, and the advantages of the random forest method cannot be reflected. If the number of decision trees in the forest exceeds a certain limit, the new decision trees will not improve the classification effect of the random forest classifier, but will drag down the overall classification speed due to the increased amount of calculation. Therefore, this paper adopts a trial and error method for the selection of k value and forest density, constantly enumerates possible k values and forest density for experiments and comparison, and finally selects the value that conforms to the dataset used in this paper.

3. Experiments

In the experimental part, NSL_KDD dataset is used, which is derived from KDD99 dataset. Although the NSL_KDD dataset is still not a perfect dataset compared with the real network environment, it is an effective benchmark dataset for intrusion detection researchers. It solves some inherent problems of KDD99 dataset, such as redundant records in the training set, so the classifier will not favor records with high frequency, and the number of records in the training set and test set is reasonable. NSL_KDD dataset is widely used in intrusion detection field. The NSL_KDD dataset contains normal and abnormal data. The data can be divided into 23 categories according to different attack modes, which can be further classified into 5 categories. In this experiment, 30% of

NSLKDD data set is used, of which 15% is used as the training set and the other 15% as the test set.

In this paper, CART decision tree based on Gini index is used to construct multiple random forests with forest density ranging from 1 to 20. The experimental results show that, with the increase of the number of decision trees in the random forest, the growth rate of experimental accuracy is decreasing. When the number of decision trees exceeds 20, the growth tends to be flat slow. Therefore, this paper divides the [1,20] interval and lists the accuracy of using PCA to reduce data to different dimensions (5 dimensions, 10 dimensions, 20 dimensions) when the number of decision trees in the random forest is 5, 10, 20 and more. The data is divided into 5 categories (normal, dos, u2r, r2l, probe) and 2 categories (normal, abnormal), That is, the classification accuracy without PCA preprocessing is compared when there is only a single decision tree. The comparison is shown in Table 1 and Table 2.

Table 1. Classification accuracy when data is divided into 5 categories

Number / Accuracy / K	**1**	**5**	10	**20**	**20+**
5	85.1	86.7	88.4	89.3	89.4
10	85.6	88.5	91.1	90.4	90.3
20	86.4	89.6	89.4	88.9	90.2
Without PCA	79.43	80.2	79.1	79.5	79.6

Table 2. Classification accuracy when data is divided into 2 categories

Number / Accuracy / K	**1**	**5**	10	**20**	**20+**
5	94.7	97.5	98.1	98.3	98.2
10	95.7	97.5	98.6	98.5	98.4
Without PCA	94.6	96.8	97.1	97.5	97.9

It can be seen from Table 1 and Table 2 that for the NSL_KDD dataset used in the experiment, the increase in the number of decision trees in the random forest has no significant impact on the accuracy. The classification accuracy rate when the data is divided into two categories is significantly higher than that when the data is divided into five categories. However, no matter whether the data is divided into 5 or 2 categories, the classification accuracy of this method is higher than that of the decision tree algorithm. When the data is divided into five categories, the accuracy can be effectively improved by using PCA to reduce the dimensions of the data and then classify them. When the data is divided into two categories, PCA dimension reduction can improve the accuracy of classification to a certain extent, but it is not as obvious as when the data is divided into five categories. Since the accuracy of PCA is not different when the data is reduced to 5 and 10 dimensions when the data is divided into 2 categories, the test is not continued when the data is reduced to 20 dimensions.

This paper also tested the accuracy of several commonly used machine learning classification methods when dividing the experimental data into two categories and five categories under the same data set, as shown in Table 3. It can be seen that the best performance is the SVM method using the kernel function, with the accuracy rate reaching 90%, and the accuracy rate of the other classification methods is only about 80%.

Table 3. Experimental results of some common classification methods in the same dataset

Classification type / method	2 categories	5 categories
SVM	82.5	78.7
SVM (kernel function)	90.8	86.3
Naive Bayesian	80.3	77.4
Logistic regression	78.6	74.9

4. Conclusion

This paper proposes an intrusion detection method based on PCA and random forest classification by using the idea of data cleaning before classification. Firstly, PCA is used to reduce the feature dimension of a large number of data to reduce the amount of computation and remove the noise in the data; Then, the reduced dimension data are trained using a random forest classifier. The experimental results show that the intrusion detection method proposed in this paper can effectively remove the noise in the data and reduce the calculation of the classifier. Compared with the common classification methods, it improves the accuracy of intrusion detection to a certain extent.

References

[1] ZHANG Shuai, DI Shaojia. Monitoring Data of Network Security in September 2019 [J]. Netinfo Security, 2019 (11): 93-94.

[2] ZHANG Lei, CUI Yong, LIU Jing. Application of Machine Learning in Cyberspace Security Research [J]. Chinese Journal of Computers, 2018, 41(09): 1943-1975.

[3] WANG Xiang, HU Xuegang, YANG Qiujie. Research on improved intrusion detection model with random forest based on feature evaluation of One-R00 [J]. Journal of He fei University of Technology (Natural Science), 2015, 38 (05): 627-630, 711.

[4] REN Jiadong, LIU Xinqian, WANG Qian. A Multi-Level Intrusion Detection Method Based on KNN Outlier Detection and Random Forests [J]. Journal of Com- puter Research and Development, 2019, 56(03): 566-575.

[5] Liang C, Li C H, Zhou L E. A PCA-BP neural network-based intrusion detection method [J]. Journal of Air Engineering University (Natural Science Edition), 2016, 17(6): 93-98.

[6] Guinde N B, Ziavras S G. Efficient hardware support for pattern matching in network intrusion detection [J]. computers & security, 2010, 29(7): 756-769.

[7] Quinlan J R. Induction of decision trees [J]. Machine learning, 1986, 1(1), 81-106.

[8] Quinlan, J. R. C4.5: Programs for Machine Learning [M]. Morgan Kaufmann Publishers, 1993: 1-12.

[9] Steinberg D. CART: classification and regression trees ［M］. The Top Ten Algorithms in Data Mining. Chapman and Hall/CRC, 2009:193-216.